

A Tutorial on Analysis and Simulation of Boolean Gene Regulatory Network Models

Yufei Xiao^{*,1,2}

¹Dept. of Epidemiology & Biostatistics; ²Greehey Children's Cancer Research Institute, University of Texas Health Science Center at San Antonio, San Antonio, TX 78229, USA

Abstract: Driven by the desire to understand genomic functions through the interactions among genes and gene products, the research in gene regulatory networks has become a heated area in genomic signal processing. Among the most studied mathematical models are Boolean networks and probabilistic Boolean networks, which are rule-based dynamic systems. This tutorial provides an introduction to the essential concepts of these two Boolean models, and presents the up-to-date analysis and simulation methods developed for them. In the Analysis section, we will show that Boolean models are Markov chains, based on which we present a Markovian steady-state analysis on attractors, and also reveal the relationship between probabilistic Boolean networks and dynamic Bayesian networks (another popular genetic network model), again *via* Markov analysis; we dedicate the last subsection to structural analysis, which opens a door to other topics such as network control. The Simulation section will start from the basic tasks of creating state transition diagrams and finding attractors, proceed to the simulation of network dynamics and obtaining the steady-state distributions, and finally come to an algorithm of generating artificial Boolean networks with prescribed attractors. The contents are arranged in a roughly logical order, such that the Markov chain analysis lays the basis for the most part of Analysis section, and also prepares the readers to the topics in Simulation section.

Received on: December 04, 2008 - Revised on: May 11, 2009 - Accepted on: May 11, 2009

1. INTRODUCTION

In most living organisms, genome carries the hereditary information that governs their life, death, and reproduction. Central to genomic functions are the coordinated interactions between genes (both the protein-coding DNA sequences and regulatory non-coding DNA sequences), RNAs and proteins, forming the so called gene regulatory networks (or genetic regulatory networks).

The urgency of understanding gene regulations from systems level has increased tremendously ever since the early stage of genomics research. A driving force is that, if we can build good gene regulatory network models and apply intervention techniques to control the genes, we may find better treatment for diseases resulting from aberrant gene regulations, such as cancer. In the past decade, the invention of high throughput technologies has made it possible to harvest large quantities of data efficiently, which is turning the quantitative study of gene regulatory networks into a reality. Such study requires the application of signal processing techniques and fast computing algorithms to process the data and interpret the results. These needs in turn have fueled the development of genomic signal processing and the use of mathematical models to describe the complex interactions between genes.

The roles of mathematical models for gene regulatory networks include:

- Describing genetic regulations at a system level;
- Enabling artificial simulation of network behavior;
- Predicting new structures and relationships;
- Making it possible to analyze or intervene in the network through signal processing methods.

Among various mathematical endeavors are two Boolean models, Boolean networks (BNs) [1] and probabilistic Boolean networks (PBNs) [2], in which each node (gene) takes on two possible values, ON or OFF (or 1 and 0), and the way genes interact with each other is formulated by standard logic functions. They constitute an important class of models for gene regulatory networks, in that they capture some fundamental characteristics of gene regulations, are conceptually simple, and their rule-based structures bear physical and biological meanings. Moreover, Boolean models can be physically implemented by electronic circuits, and demonstrate rich dynamics that can be studied using mathematical and signal processing theory (for instance, Markov chains [2, 3]).

In practice, Boolean models have been successfully applied to describe real gene regulatory relations (for instance, the *drosophila* segment polarity network [4]), and the attractors of BNs and PBNs have been associated with cellular phenotypes in the living organisms [5]. The association of network attractors and actual phenotypes has inspired the development of control strategy [6] to increase the possibility of reaching desirable attractors ("good"

*Address correspondence to this author at the Computational Biology and Bioinformatics Division, Greehey Children's Cancer Research Institute, University of Texas Health Science Center at San Antonio, San Antonio, TX 78229, USA; Tel: 210-562-9080; Fax: 210-562-9135; E-mail: xiaoy4@uthscsa.edu

phenotypes) and decrease the likelihood of undesirable attractors (“bad” phenotypes such as cancer). The effort of applying control theory to Boolean models is especially appealing in the medical community, as it holds potential to guide the effective intervention and treatment in cancer.

The author would like to bring the fundamentals of Boolean models to a wider audience in light of their theoretical value and pragmatic utility. This tutorial will introduce the basic concepts of Boolean networks and probabilistic Boolean networks, present the mathematical essentials, and discuss some analyses developed for the models and the common simulation issues. It is written for researchers in the genomic signal processing area, as well as researchers with general mathematics, statistics, engineering, or computer science backgrounds who are interested in this topic. It intends to provide a quick reference to the fundamentals of Boolean models, allowing the readers to apply those techniques to their own studies. Formal definitions and mathematical foundations will be laid out concisely, with some in-depth mathematical details left to the references.

2. PRELIMINARIES

In Boolean models, each variable (known as a *node*) can take two possible values, 1 (ON) and 0 (OFF). A node can represent a gene, RNA sequence, or protein, and its value (1 or 0) indicates its measured abundance (expressed or unexpressed; high or low). In this paper, we use “node” and “gene” interchangeably.

A *state* in Boolean models is a binary vector of all the gene values measured at the same time, and is also called the gene activity (or expression) profile (GAP). The *state space* of a Boolean model consists of all the possible states, and its size will be 2^n for a model with n nodes.

Definition 1 [2, 7] A Boolean network is defined on a set of n binary-valued nodes (genes) $V = \{x_1, \dots, x_n\}, x_i \in \{0,1\}$, where each node x_i has k_i parent nodes (regulators) chosen from V , and its value at time $t + 1$ is determined by its parent nodes at t through a Boolean function f_i ,

$$x_i(t+1) = f_i(x_{i_1}(t), x_{i_2}(t), \dots, x_{i_{k_i}}(t)), \quad \{i_1, \dots, i_{k_i}\} \subseteq \{1, \dots, n\}. \quad (1)$$

k_i is called the *connectivity* of x_i , and f_i is the regulatory function. Defining network function $\mathbf{f} = (f_1, \dots, f_n)$, we denote the Boolean network as $\beta(V, \mathbf{f})$. Let the network state at time t be $\mathbf{x}(t) = (x_1(t), \dots, x_n(t))$, the state transition $\mathbf{x}(t) \rightarrow \mathbf{x}(t + 1)$ is governed by \mathbf{f} , written as $\mathbf{x}(t + 1) = \mathbf{f}(\mathbf{x}(t))$.

In Boolean networks, genetic interactions and regulations are hard-wired with the assumption of biological determinism. However, any gene regulatory network is not a closed system and has interactions with its environment and other genetic networks, and it is also likely that genetic regulations are inherently stochastic; therefore, Boolean networks will have limitations in their modeling power.

Probabilistic Boolean networks were introduced to address this issue [2, 7], such that they are composed of a family of Boolean networks, each of which is considered a context [8]. At any given time, gene regulations are governed by one component Boolean network, and network switchings are possible such that at a later time instant, genes can interact under a different context. In this sense, probabilistic Boolean networks are more flexible in modeling and interpreting biological data.

Definition 2 [2, 3, 7] A probabilistic Boolean network is defined on $V = \{x_1, \dots, x_n\}, x_i \in \{0,1\}$, and consists of r Boolean networks $\beta_1(V, \mathbf{f}_1), \dots, \beta_r(V, \mathbf{f}_r)$, with associated network selection probabilities c_1, \dots, c_r such that $\sum_{j=1}^r c_j = 1$. The network function of the j -th BN is $\mathbf{f}_j = (f_j^{(1)}, \dots, f_j^{(n)})$. At any time, genes are regulated by one of the BNs, and at the next time instant, there is a probability q (switching probability) to change network; once a change is decided upon, we choose a BN randomly (from r BNs) by the selection probabilities. Let p be the rate of random gene perturbation (flipping a gene value from 0 to 1 or 1 to 0), the state transition of PBN at t (assuming operation under β_j) is probabilistic, namely [3],

$$\mathbf{x}(t+1) = \begin{cases} \mathbf{f}_j(\mathbf{x}(t)), & \text{with probability } (1-p)^n, \\ \mathbf{x}(t) \oplus \gamma, & \text{with probability } 1 - (1-p)^n, \end{cases} \quad (2)$$

where \oplus is bit-wise modulo-2 addition, $\gamma = (\gamma_1, \dots, \gamma_n)$ is a random vector with $\Pr\{\gamma_i = 1\} = p$, and $\mathbf{x}(t) \oplus \gamma$ denotes a random perturbation on the state $\mathbf{x}(t)$ (one or more genes are flipped). Let the set of network functions be $\mathbf{F} = \{\mathbf{f}_1, \dots, \mathbf{f}_r\}$, and we denote the PBN by $G(V, \mathbf{F}, \mathbf{c}, p)$ (see Remark 1).

Alternatively, the PBN can be represented as $G(V, \Psi, \alpha, p)$, with $\Psi = \{\Psi_1, \dots, \Psi_n\}$ and $\alpha = \{\alpha_1, \dots, \alpha_n\}$. In this representation, each node x_i is regarded as being regulated by a set of $l(i)$ Boolean functions $\Psi_i = \{\psi_{i_1}^{(i)}, \dots, \psi_{i_{l(i)}}^{(i)}\}$ with the corresponding set of function selection probabilities $\alpha_i = \{\alpha_{i_1}^{(i)}, \dots, \alpha_{i_{l(i)}}^{(i)}\}$ ($\sum_{j=1}^{l(i)} \alpha_j^{(i)} = 1$). The two representations are related such that any network function \mathbf{f}_j is a realization of the regulatory functions of n genes by choosing one function from the function set Ψ_i for each gene x_i , and we can write

$$\mathbf{f}_j = (\psi_{j_1}^{(1)}, \dots, \psi_{j_n}^{(n)}), \quad j_i \in \{1, \dots, l(i)\}. \quad (3)$$

Moreover, if it is an independent PBN, namely $\Pr\{\psi_{j_1}^{(1)}, \dots, \psi_{j_n}^{(n)}\} = \prod_{i=1}^n \Pr\{\psi_{j_i}^{(i)}\}$, \mathbf{c} and α are related by

$$c_j = \prod_{i=1}^n \alpha_{j_i}^{(i)}. \quad (4)$$

Remark 1 q does not appear in the PBN representation, because according to the network switching scheme described, it can be shown that the probability of being in the β_j at any time is equal to c_j , regardless of q . However, if we modify the network switching scheme such that, once a network switch is decided upon, we randomly choose any network *other than* the current network, it will require the definition of $r(r-1)$ conditional selection probabilities, $c_{jk} = \Pr\{\mathbf{f}_k | \mathbf{f}_j\}$, $k, j \in \{1, \dots, r\}, k \neq j$, and the derivation of $\Pr\{\mathbf{f}_j\}$ (the probability of being in β_j) is left as an exercise to the reader.

A Boolean model with finite number of nodes has a finite state space. From the definition of Boolean network, it follows that its state transitions are deterministic, that is, given a state, its successor state is unique. Naturally, if we represent the whole state space and the transitions among the states of a BN graphically, we can have a *state transition diagram*.

Definition 3 The state transition diagram of an n -node Boolean network $\beta(V, \mathbf{f})$ is a directed graph $D(S, E)$. S is a set of 2^n vertices, each representing a possible state of a Boolean network; E is a set of 2^n edges, each pointing from a state to its successor state in state transition. If a state transits to itself, then the edge is a loop. The state transitions are computed by evaluating $\mathbf{x}(t+1) = \mathbf{f}(\mathbf{x}(t))$ exactly 2^n times, each time $\mathbf{x}(t)$ being $00\dots0, 00\dots1, \dots, 11\dots1$ respectively.

Fig. (1) is an example of state transition diagram of a three-node BN. Like BNs, a PBN also has finite state space. Although state transitions in a PBN are not deterministic, they can be represented probabilistically. We will show how to construct the state transition diagram of a PBN in the Simulation section.

With the help of state transition diagram, such as the one in Fig. (1), we can easily visualize that in a BN, any state trajectory in time $\mathbf{x}(0) \rightarrow \mathbf{x}(1) \rightarrow \mathbf{x}(2) \rightarrow \dots$ must end up in

a “trap”, and stay there forever unless a gene perturbation occurs. Similarly, if neither gene perturbation nor network switching has occurred, a time trajectory in a PBN will end up in a “trap” in one of the component BNs too; however, either gene perturbation or a network switch may cause it to escape from the trap. In spite of this, when gene perturbation and network switching are rare, a PBN is most likely to reach a “trap” before either occurs and will spend a reasonably long time there.

Definition 4 Starting from any initial state in a finite Boolean network, when free of gene perturbation, state transitions will allow the network to reach a finite set of states $\{\mathbf{a}_1, \dots, \mathbf{a}_m\}$ and cycle among them in a fixed order forever. The set of states is called an attractor, denoted by A . If A contains merely one state, it is a singleton attractor; otherwise, it is an attractor cycle. The set of states from which the network will eventually reach an attractor A constitutes the basin of attraction of A . A BN may have more than one attractor.

The attractors of a PBN are defined as the union of attractors of its component BNs. In particular, if a PBN is composed of r BNs, and the k -th BN has m_k attractors, $A_{k1}, A_{k2}, \dots, A_{km_k}$, then the attractors of PBN are $\{A_{11}, A_{12}, \dots, A_{1m_1}\} \cup \dots \cup \{A_{r1}, A_{r2}, \dots, A_{rm_r}\}$.

In a BN, different basins of attraction are depicted in the state transition diagram as disjoint subgraphs. In Fig. (1), $D(S, E)$ is composed of three disjoint subgraphs, $D_1(S_1, E_1)$, $D_2(S_2, E_2)$, and $D_3(S_3, E_3)$. 110 and 101 are singleton attractors, while 100 and 111 constitute a cycle. Their respective basins of attraction are $S_1 = \{000, 010, 110\}$, $S_2 = \{101\}$ and $S_3 = \{001, 011, 100, 111\}$.

We are interested in the attractors of a Boolean model for at least two reasons: (1) Attractors represent the stable states of a dynamic system, thus they are tied to the long term behavior of Boolean models; (2) Earlier researchers demonstrated the association of cellular phenotype with

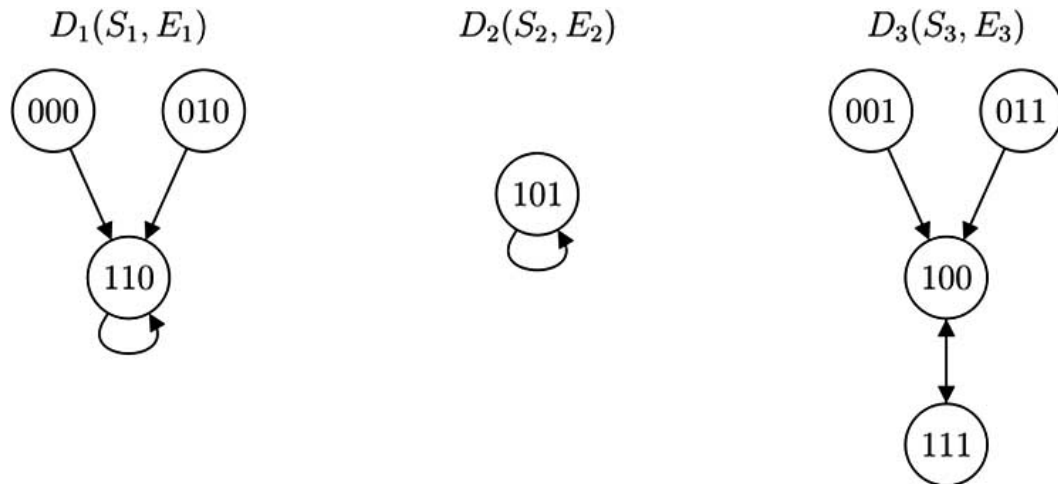


Fig. (1). A state transition diagram $D(S, E)$.

attractors [5], thus giving a biological meaning to the attractors. Intuitively, when an attractor has a large basin of attraction, the corresponding phenotype is more likely than that of an attractor with much smaller basin of attraction. To develop intervention strategies that change the long term behavior of Boolean models, it is important to study the attractors.

3. ANALYSES OF BOOLEAN MODELS

Although analysis and simulation are two parallel subjects with Boolean models, the former includes some essential results that lay a foundation for the latter. In this section, we visit Boolean model analysis first.

One of the central ideas with Boolean models is their connection with Markov chains (subsection 3.1). Because of this, Boolean models, under certain conditions, possess steady-state distributions. The steady-state probabilities of attractors, which indicate the long-run trend of network dynamics, can be found analytically *via* Markov chain analysis (subsection 3.2). Moreover, the relationship between PBNs and Bayesian networks (another class of gene regulatory network models) can be established in a similar manner (subsection 3.3). Lastly, a subsection will be dedicated to structural analysis, which opens a door to other topics beyond this tutorial (such as control of genetic networks).

3.1. Markov Chain Analysis

As readers will find out soon, the transition probability matrix introduced below is not only a convenience in Markov chain analysis, but also finds itself useful in simulation, to be discussed in Section 4.

3.1.1. Transition Probability Matrix

On a Boolean model of n nodes, a *transition probability matrix* $T = [t_{ij}]_{2^n \times 2^n}$ can be defined where t_{ij} indicates the probability of transition from one state (which is equal to $i-1$ if we convert the binary vector to an integer) to another state (which corresponds to $j-1$).

In a Boolean network $\beta(V, \mathbf{f})$, t_{ij} can be computed by

$$t_{ij} = \begin{cases} 1, & \exists \mathbf{s} \in \{0,1\}^n \text{ such that } \text{dec}(\mathbf{s}) = i-1, \text{dec}(\mathbf{f}(\mathbf{s})) = j-1, \\ 0, & \text{otherwise,} \end{cases} \quad (5)$$

where $\text{dec}(\cdot)$ converts a binary vector to an integer, for instance, $\text{dec}(00101) = 5$. Since BN is deterministic, T contains one 1 on each row, and all other elements are 0's.

In a PBN consisting of r BNs $\beta_1(V, \mathbf{f}_1), \dots, \beta_r(V, \mathbf{f}_r)$, t_{ij} can be computed as follows [2, 3]. Note that p (random gene perturbation rate) and γ are defined as in Definition 2, and c_k is the selection probability of β_k .

$$t_{ij} = \sum_{k=1}^r \Pr \{ \beta_k \text{ is selected} \} \cdot$$

$$\begin{aligned} & \Pr \{ \mathbf{s} \rightarrow \mathbf{w}, \text{dec}(\mathbf{s}) = i-1, \text{dec}(\mathbf{w}) = j-1 | \beta_k \text{ is selected} \} \\ &= \sum_{k=1}^r c_k \cdot [\Pr \{ \mathbf{s} \rightarrow \mathbf{w} \text{ by state transition, } \text{dec}(\mathbf{s}) = i-1, \text{dec}(\mathbf{w}) = j-1 | \mathbf{f}_k \} \\ &+ \Pr \{ \mathbf{s} \rightarrow \mathbf{w} \text{ by random gene perturbation, } \text{dec}(\mathbf{s}) = i-1, \text{dec}(\mathbf{w}) = j-1 | \mathbf{f}_k \}] \\ &= \sum_{k=1}^r c_k \cdot [(1-p)^n \Pr \{ \mathbf{s} \rightarrow \mathbf{f}_k(\mathbf{s}), \text{dec}(\mathbf{s}) = i-1, \text{dec}(\mathbf{f}_k(\mathbf{s})) = j-1 | \mathbf{f}_k \} \\ &+ \Pr \{ \mathbf{s} \rightarrow \mathbf{s} \oplus \gamma, \gamma \neq (0, \dots, 0), \text{dec}(\mathbf{s}) = i-1, \text{dec}(\mathbf{s} \oplus \gamma) = j-1 | \mathbf{f}_k \}] \\ &= \sum_{k=1}^r c_k \cdot [(1-p)^n \mathbf{1}_{\{\mathbf{s} \rightarrow \mathbf{f}_k(\mathbf{s}), \text{dec}(\mathbf{s}) = i-1, \text{dec}(\mathbf{f}_k(\mathbf{s})) = j-1\}} + p_\gamma \mathbf{1}_{\{i \neq j\}}], \quad (6) \end{aligned}$$

where $\mathbf{1}$'s are indicator functions, $p_\gamma = \binom{n}{l} p^l (1-p)^{n-l}$, l = number of 1's in the random vector $\gamma = (\gamma_1, \dots, \gamma_n)$, and l indicates the Hamming distance between \mathbf{s} and \mathbf{w} .

When taking a closer look at Eq. (6), we find that T is the sum of a fixed transition matrix \bar{T} and a perturbation matrix \tilde{T} ,

$$\bar{T} = (1-p)^n \sum_{j=1}^r c_j T_j, \quad (7)$$

$$\tilde{T} = [\tilde{t}_{ij}], \tilde{t}_{ij} = n \binom{n}{n_{ij}} p^{n_{ij}} (1-p)^{n-n_{ij}} \mathbf{1}_{\{i \neq j\}}, \quad (8)$$

where T_j and c_j are the transition probability matrix and the network selection probability of the j -th Boolean network, respectively; n_{ij} is the Hamming distance between states \mathbf{s} and \mathbf{w} , with $\text{dec}(\mathbf{s}) = i-1$ and $\text{dec}(\mathbf{w}) = j-1$.

T_j is sparse with only 2^n non-zero entries (out of $2^n \times 2^n$ entries), where each is a state transition driven by the network function \mathbf{f}_j and involves n computations. \tilde{T} depends only on n and p , and involves n computations. Thus, the computational complexity for T is $O(n \cdot r \cdot 2^n)$ [9].

3.1.2. Boolean Models are Markov Chains

Given the definition of T matrix in Section 3.1.1, we can see that a Boolean model with n genes is a homogeneous Markov chain of $N = 2^n$ states, with T being the Markov matrix and $\sum_{j=1}^{2^n} t_{ij} = 1, \forall i$. A state \mathbf{x} (a binary vector of length n) in Boolean model has one-to-one correspondence with the i -th state ($1 \leq i \leq N$) in the associated Markov chain by $\text{dec}(\mathbf{x}) = i-1$.

What is the use of matrix T ? Let $W = T^n$, we can show that the (i, j) -th element of W is equal to the probability of transition from the i -th state to the j -th state of the Markov chain in n steps,

$$w_{ij} = \Pr\{\mathbf{x}(t+n) = \mathbf{z}, \text{dec}(\mathbf{z}) = j-1 \mid \mathbf{x}(t) = \mathbf{y}, \text{dec}(\mathbf{y}) = i-1\}.$$

The proof is left as an exercise to the reader.

An N -state Markov chain possesses a *stationary distribution* (or invariant distribution) if there exists a probability distribution $\pi = (\pi_1, \dots, \pi_N)$ such that

$$\pi = \pi T.$$

$\pi = \pi T$ implies $\pi = \pi T^n, \forall n$. Thus in a Markov chain with stationary distribution π , if we start from the i -th state with probability π_i , the chance of being in any state j after an arbitrary number of steps is always π_j .

An N -state Markov chain possesses a *steady-state distribution* $\pi^* = (\pi_1^*, \dots, \pi_N^*)$ if starting from any initial distribution π

$$\pi^* = \lim_{k \rightarrow \infty} \pi T^k,$$

it means that regardless of the initial state, the probability of a Markov chain being in the i -th state in the long run is π_i^* .

A Markov chain possessing a stationary distribution does not necessarily possess a steady-state distribution.

Why should it be of our concern if the Markov chain has a steady-state distribution or not? This is because we are interested in the Boolean model associated with the Markov chain, and would like to know how it behaves in the long-run. As a reminder, the attractors of a Boolean model are often associated with cellular phenotype, and by finding out the steady-state probabilities of a given attractor, we can have a general picture of the likelihood of a certain phenotype. When a Boolean model possesses (namely, its Markov chain possesses) a steady-state distribution, we can find those probabilities by simulating the model for a long time, starting from an arbitrary initial state $\mathbf{x}(0)$. In fact, this implies the equivalence of “space average” and “time average”, as is a common concept in stochastic processes.

When will a Markov chain possess a steady-state distribution? It turns out that an *ergodic* Markov chain will do. A Markov chain is said to be ergodic if it is irreducible and aperiodic [10].

Definition 5 A Markov chain is irreducible if it is possible to go from every state to every state (not necessarily in one move).

Definition 6 In a Markov chain, a state has period d if starting from this state, we can only return to it in n steps and n is a multiple of d . A state is periodic if it has some period > 1 . A Markov chain is aperiodic if none of its state is periodic.

A Boolean network possesses a stationary distribution, but not a steady-state distribution unless it has one singleton attractor and no other attractors. Here we show how to find a stationary distribution. Assume a BN has m singleton

attractors, $\mathbf{a}_1, \dots, \mathbf{a}_m$, or an attractor cycle $\{\mathbf{a}_1, \dots, \mathbf{a}_m\}$, where $\text{dec}(\mathbf{a}_1) = i_1 - 1, \dots, \text{dec}(\mathbf{a}_m) = i_m - 1$, then π with $\pi_{i_1} = \dots = \pi_{i_m} = 1/m$ and $\pi_j = 0, j \notin \{i_1, \dots, i_m\}$ is a stationary distribution (the proof is left as an exercise to the reader). If a BN has a combination of singleton attractors and cycles, π can be constructed such that the probabilities corresponding to the singleton attractors are equal, the probabilities corresponding to the states within each attractor cycle are equal, and $\sum_{i=1}^N \pi_i = 1$. When there is only one attractor in the BN, the stationary distribution is unique.

When $p, q > 0$, a PBN possesses steady-state distribution, because the Markov chain corresponding to the PBN is ergodic. Interested readers can find the proof in [3]. Now that PBN has a steady-state distribution, we can obtain such distribution in two ways: (2) solving the linear equations $\pi(T - I) = 0, \sum_{i=1}^N \pi_i = 1$ (I is the identity matrix), and interested readers can consult books on linear algebra; (2) using the empirical methods in Section 4.3. If we are interested in the steady-state probabilities of the attractors only, an analytic method exists, to be discussed next.

3.2. Analytic Method for Computing the Steady-State Probabilities of Attractors

Recall from Section 2 that attractors are important to the long-term behavior of Boolean models because they are associated with cellular phenotypes; now we also know that PBNs possess steady-state distributions, which means that a PBN has a unique long-term trend independent of initial state. Therefore, we would naturally ask the question, how can we find the long-term probabilities of these attractors which are so important to us?

In the following, we will present a Markov chain based analytic method that answers this question, and more details, including proofs, can be found in [11].

3.2.1. Steady-State Distributions of Attractors in a BN with Perturbations

First consider a special case of PBN, Boolean network with perturbations (BNp), in which any gene has a probability p of flipping its value. A BNp inherits all the attractors and corresponding basins of attraction from the original BN. Because of the random gene perturbations, BNps possess steady-state distributions (the proof is similar to that of PBN, and it is left as an exercise to the reader).

A BNp defined on $V = \{x_1, \dots, x_n\}$ with gene perturbation rate p can be viewed as homogenous irreducible Markov chain \mathbf{X}_t with state space $\{0,1\}^n$. Let $\mathbf{x}, \mathbf{y} \in \{0,1\}^n$ be any two states, then at any time t , $P_{\mathbf{y}}(\mathbf{x}) = \Pr\{\mathbf{X}_{t+1} = \mathbf{x} \mid \mathbf{X}_t = \mathbf{y}\}$ is the probability of state transition from \mathbf{y} to \mathbf{x} .

For \mathbf{X}_t , there exists a unique steady-state distribution π . Let the steady-state probability of state \mathbf{x} be $\pi(\mathbf{x})$, and let

$B \subset \{0,1\}^n$ be a collection of states, then the steady-state probability of B is $\pi(B) = \sum_{\mathbf{x} \in B} \pi(\mathbf{x})$.

Assume the BNp has attractors A_1, \dots, A_m , with corresponding basins of attraction (or simply referred to as basins) B_1, \dots, B_m . Since the attractors are subsets of the basins,

$$\pi(A_k) = \pi(A_k | B_k) \pi(B_k). \tag{9}$$

Therefore, we can compute the steady-state probability of any attractor A_k by the following two steps: (1) the steady-state probability of basin B_k , $\pi(B_k)$, and (2) the conditional probability of attractor A_k given its being in B_k , $\pi(A_k | B_k)$.

(I). Obtaining the Steady-State Probability of Basin, $\pi(B_k)$

Define a random variable $\tau(t)$ which measures the time elapsed between the last perturbation and the current time t . $\tau(t) = 0$ means a perturbation occurs at t . For any starting state \mathbf{h} , let

$$P_{B_i}^*(B_k) = \lim_{t \rightarrow \infty} \Pr\{\mathbf{X}_t \in B_k | \mathbf{X}_{t-1} \in B_i, \mathbf{X}_0 = \mathbf{h}, \tau(t) = 0\}, \tag{10}$$

and define the conditional probability of being in state $\mathbf{x} \in B$ given that the system is inside a set B , prior to a perturbation,

$$\pi^*(\mathbf{x} | B) := \lim_{t \rightarrow \infty} \Pr\{\mathbf{X}_{t-1} = \mathbf{x} | \mathbf{X}_{t-1} \in B, \mathbf{X}_0 = \mathbf{h}, \tau(t) = 0\}. \tag{11}$$

The following theorem represents the steady-state distribution of the basins as the solution of a group of linear equations, where the coefficients are $P_{B_i}^*(B_k)$'s. The lemma that follows gives the formula for the coefficients.

Theorem 1

$$\pi(B_k) = \sum_{i=1}^m P_{B_i}^*(B_k) \pi(B_i). \tag{12}$$

Lemma 1

$$P_{B_i}^*(B_k) = \sum_{\mathbf{x} \in B_k} \sum_{\mathbf{y} \in B_i} P_{\mathbf{y}}^*(\mathbf{x}) \pi^*(\mathbf{y} | B_i), \tag{13}$$

where $P_{\mathbf{y}}^*(\mathbf{x})$ is the probability that state transition goes from \mathbf{y} to \mathbf{x} in one step by gene perturbation.

Now the only unknown is $\pi^*(\mathbf{y} | B_i)$. When p is small, the system spends majority of the time inside an attractor, and we can use the following approximation,

$$\pi^*(\mathbf{y} | B_i) \approx \frac{1}{|A_i|} \mathbf{1}_{[\mathbf{y} \in A_i]}, \tag{14}$$

where $|A_i|$ is the cardinality of A_i . Therefore,

$$P_{B_i}^*(B_k) \approx \frac{1}{|A_i|} \sum_{\mathbf{x} \in B_k} \sum_{\mathbf{y} \in A_i} P_{\mathbf{y}}^*(\mathbf{x}). \tag{15}$$

(II). Obtaining the Steady-State Probability of Attractor, $\pi(A_k)$

Lemma 2 For basin B_k , initial state \mathbf{h} , and fixed value $j \geq 0$,

$$\lim_{t \rightarrow \infty} \Pr\{\mathbf{X}_{t-j} = \mathbf{x} | \mathbf{X}_{t-j} \in B_k, \mathbf{X}_0 = \mathbf{h}, \tau(t) = j\} = \frac{1}{\pi(B_k)} \sum_{i=1}^m \sum_{\mathbf{y} \in B_i} P_{\mathbf{y}}^*(\mathbf{x}) \pi^*(\mathbf{y} | B_i) \pi(B_i). \tag{16}$$

Lemma 3 If $\delta(\mathbf{x}, A_k)$ is the number of iterations of \mathbf{f} needed to reach the attractor A_k from the state \mathbf{x} , then for any $\mathbf{x} \in A_k$, $b < 1$,

$$\sum_{j=\delta(\mathbf{x}, A_k)}^{\infty} (1-b)b^j = b^{\delta(\mathbf{x}, A_k)}. \tag{17}$$

Applying the two lemmas and letting $b = (1-p)^n$, we can obtain the steady-state probability of attractor A_k .

Theorem 2

$$\pi(A_k) = \sum_{i=1}^m \left[\sum_{\mathbf{x} \in B_k} \sum_{\mathbf{y} \in B_i} P_{\mathbf{y}}^*(\mathbf{x}) \pi^*(\mathbf{y} | B_i) (1-p)^{n\delta(\mathbf{x}, A_k)} \right] \pi(B_i). \tag{18}$$

When p is small, using the approximation in Eq. (14), we have

$$\pi(A_k) \approx \sum_{i=1}^m \frac{1}{|A_i|} \left[\sum_{\mathbf{x} \in B_k} \sum_{\mathbf{y} \in B_i} P_{\mathbf{y}}^*(\mathbf{x}) (1-p)^{n\delta(\mathbf{x}, A_k)} \right] \pi(B_i). \tag{19}$$

3.2.2. Steady-State Distributions of Attractors in a PBN

In a PBN, we represent the pair (\mathbf{x}, \mathbf{f}) as the state of a homogeneous Markov chain, $(\mathbf{X}_t, \mathbf{F}_t)$, and the transition probabilities are defined as

$$P_{\mathbf{y}, \mathbf{g}}(\mathbf{x}, \mathbf{f}) = \Pr\{\mathbf{X}_{t+1} = \mathbf{x}, \mathbf{F}_{t+1} = \mathbf{f} | \mathbf{X}_t = \mathbf{y}, \mathbf{F}_t = \mathbf{g}\} \tag{20}$$

Assume the PBN is composed of r BNs $\beta_1(V, \mathbf{f}_1), \dots, \beta_r(V, \mathbf{f}_r)$. Within BN β_k , the attractors and basins are denoted A_{ki} and B_{ki} , $i = 1, \dots, m_k$. The computation of the steady-state probabilities are now split into three steps: (1) steady-state probabilities $\pi(B_{ki}, \mathbf{f}_k)$ of the basins, (2) conditional probabilities $\pi(A_{ki}, \mathbf{f}_k | B_{ki}, \mathbf{f}_k)$, and (3) approximation to the marginal steady-state probabilities $\pi(A_{ki})$ (since different BNs may have the same attractor).

The computations in steps (1) and (2) are similar to that of BNp, with (B_{ki}, \mathbf{f}_k) in place of B_k whenever applicable, and there is one extra summation $\sum_{k=1}^r$ for the r component BNs. Interested readers can find details in [11].

From steps (1) and (2), we can obtain $\pi(A_{ki}, \mathbf{f}_k)$. The last step sums up $\pi(A_{ki}, \mathbf{f}_l)$ over l whenever the l -th BN has A_{ki} as an attractor,

$$\pi(A_{ki}) = \sum_{l=1}^r \pi(A_{ki}, \mathbf{f}_l). \quad (21)$$

Since $\pi(A_{ki}, \mathbf{f}_l)$ is unknown when $k \neq l$, we use the following approximation when p is small,

$$\pi(A_{ki}, \mathbf{f}_l | A_{lj}, \mathbf{f}_l) \approx \frac{|A_{ki} \cap A_{lj}|}{|A_{lj}|}. \quad (22)$$

Thus,

$$\pi(A_{ki}, \mathbf{f}_l) \approx \sum_{j=1}^{m_l} \pi(A_{ki}, \mathbf{f}_l | A_{lj}, \mathbf{f}_l) \pi(A_{lj}, \mathbf{f}_l) \approx \sum_{j=1}^{m_l} \frac{|A_{ki} \cap A_{lj}|}{|A_{lj}|} \pi(A_{lj}, \mathbf{f}_l), \quad (23)$$

and

$$\pi(A_{ki}) \approx \sum_{l=1}^r \sum_{j=1}^{m_l} \frac{|A_{ki} \cap A_{lj}|}{|A_{lj}|} \pi(A_{lj}, \mathbf{f}_l). \quad (24)$$

3.3. Relationship Between PBNs and Bayesian Networks

Bayesian networks (BaN) are graphic models that describe the conditional probabilistic dependencies between variables, and have been used to model genetic regulatory networks [12]. An advantage of BaNs is that they involve model selection to optimally explain the observed data [2]; BaNs can use either continuous or discrete variables, which is more flexible for modeling. In comparison, Boolean models have explicit regulatory rules that carry biological information, which can be more appealing to biologist than the statistic representation of BaNs. Although Boolean models use binary-quantized variables which sets a limitation on the data usage, they are computational less complex than BaNs when learning the network structure from data (see Section 3.3 of [2] for a more detailed discussion and references). Since network structure learning is out of scope of this article, interested readers can refer to [12] for Bayesian learning, [13] for Boolean network learning, and [8, 14] for PBN learning.

While BNs are deterministic, PBNs and BaNs are related by their probabilistic nature; like PBNs, dynamic BaNs can be considered as Markov chains too. In the following analysis, we will show that equivalence between PBNs and BaNs can be established under certain conditions [15]. In this analysis, the random gene perturbation rate p in PBN is assumed to be 0.

A BaN with n random variables X_1, \dots, X_n (not necessarily binary) is represented by $Ba(H, \Theta)$, where H is a directed acyclic graph whose vertices correspond to the n variables and Θ is a set of conditional probability distributions induced by graph H . Letting $\mathbf{X} = (X_1, \dots, X_n)$, x_i be a realization of the random variable X_i , and $\text{Pa}(X_i)$

be the parents of X_i , the unique joint probability distribution over the n variables is given by

$$\Pr\{x_1, \dots, x_n\} = \prod_{i=1}^n \Pr\{x_i | \text{Pa}(X_i)\}.$$

A dynamic Bayesian network (DBN) is a temporal extension of BaN, and consists of two parts: (1) an initial BaN $Ba_0 = (H_0, \Theta_0)$ that defines the joint distribution of the variables $x_1(0), \dots, x_n(0)$, and (2) a transition BaN $Ba_1 = (H_1, \Theta_1)$ that defines the transition probabilities $\Pr\{\mathbf{X}(t) | \mathbf{X}(t-1)\}$, $\forall t$. Let \mathbf{X} represent a realization of \mathbf{X} , and the joint distribution of $\mathbf{X}(0), \dots, \mathbf{X}(T)$ can be expressed by

$$\Pr\{\mathbf{x}(0), \dots, \mathbf{x}(T)\} = \Pr\{\mathbf{x}(0)\} \prod_{t=1}^T \Pr\{\mathbf{x}(t) | \mathbf{x}(t-1)\} \\ = \prod_{i=1}^n \Pr\{x_i(0) | \text{Pa}(X_i(0))\} \cdot \prod_{t=1}^T \prod_{j=1}^n \Pr\{x_j(t) | \text{Pa}(X_j(t))\}. \quad (25)$$

In a PBN $G(V, \mathbf{F}, \mathbf{c})$, where $V = \{x_1, \dots, x_n\}$, $x_i \in \{0, 1\}$ and $\mathbf{F} = \{\mathbf{f}_1, \dots, \mathbf{f}_r\}$, the joint probability distribution of states over the time period $[0, T]$ can be expressed as

$$\Pr\{\mathbf{x}(0), \dots, \mathbf{x}(T)\} = \Pr\{\mathbf{x}(0)\} \prod_{t=1}^T \Pr\{\mathbf{x}(t-1) \rightarrow \mathbf{x}(t)\}.$$

For an independent PBN,

$$\Pr\{\mathbf{x}(0), \dots, \mathbf{x}(N)\} = \Pr\{\mathbf{x}(0)\} \prod_{t=1}^T \prod_{i=1}^n \Pr\{\mathbf{x}(t-1) \rightarrow x_i(t)\}. \quad (26)$$

3.3.1. An Independent PBN as a Binary-Valued DBN

Let the independent PBN be $G(V, \Psi, \alpha)$ (the alternative representation, see what follows Definition 2). First, since a BaN can represent arbitrary joint distribution, the distribution of the initial state of PBN, $\Pr\{\mathbf{x}_0\}$, can be represented by some Ba_0 . Second, to construct $Ba_1(H_1, \Theta_1)$ from the PBN, we let set $\mathbf{X}_j^{(i)} \subseteq V$ denote the regulators of gene x_i in function $\psi_j^{(i)}$,

$$\text{Pa}(x_i) = \cup_{j=1}^{l(i)} \mathbf{X}_j^{(i)}. \quad (27)$$

We construct graph H_1 such that there are two layers of nodes, the first layer has nodes $X_1(t-1), \dots, X_n(t-1)$, the second layer has nodes $X_1(t), \dots, X_n(t)$, and there exists a directed edge from $X_k(t-1)$ to $X_i(t)$ if $\exists j \in \{1, \dots, l(i)\}$ in the PBN such that $x_k \in \mathbf{X}_j^{(i)}$. Thus in H_1 , $\text{Pa}(X_i(t))$ corresponds to the set of all possible regulators of x_i in the PBN.

Let D_i be the joint distribution of the variables in $\text{Pa}(x_i)$, and recall that $\alpha_j^{(i)} = \Pr\{\psi_j^{(i)} \text{ is used}\}$, then

$$\Pr\{X_i = 1\} = \sum_{j=1}^{l(i)} \Pr\{X_i = 1 \mid \psi_j^{(i)} \text{ is used}\} \cdot \alpha_j^{(i)} \quad (28)$$

$$= \sum_{j=1}^{l(i)} \left[\sum_{\mathbf{x} \in \{0,1\}^{|\text{Pa}(X_i)|}} D_i(\mathbf{x}) \psi_j^{(i)}(\mathbf{x}) \right] \cdot \alpha_j^{(i)} \quad (29)$$

$$= \sum_{\mathbf{x} \in \{0,1\}^{|\text{Pa}(X_i)|}} D_i(\mathbf{x}) \left[\sum_{j=1}^{l(i)} \psi_j^{(i)}(\mathbf{x}) \alpha_j^{(i)} \right], \quad (30)$$

and we have

$$\Pr\{X_i(t) = 1 \mid \text{Pa}(X_i(t)) = \mathbf{z}\} = \sum_{j=1}^{l(i)} \psi_j^{(i)}(\mathbf{z}) \alpha_j^{(i)}. \quad (31)$$

Eq. (31) defines Θ_1 (induced by H_1) for each node, thus any independent PBN $G(V, \Psi, \alpha)$ can be expressed as a binary DBN (Ba_0, Ba_1) .

Remark 2 Strictly speaking, the input variables for $\psi_j^{(i)}$ are a subset of $\text{Pa}(x_i)$, so the notations in Eqs. (29-31) are not accurate when we use the same vector \mathbf{x} (or \mathbf{z}) for $\psi_j^{(i)}$ and for D_i (or $\text{Pa}(X_i(t))$). We should understand that those notations are only used as a convenience.

3.3.2. A Binary-Valued DBN as an Independent PBN

Assume DBN (Ba_0, Ba_1) defined on $\mathbf{X} = (X_1, \dots, X_n)$ is given, and X_i 's are binary-valued random variables. Now we demonstrate how to construct a PBN. Define the set of nodes $V = \{x_1, \dots, x_n\}$ in PBN corresponding to X_1, \dots, X_n , and let the distribution of PBN initial state $\mathbf{x}_0 = (x_1(0), \dots, x_n(0))$ match Θ_0 in $Ba_0(H_0, \Theta_0)$.

In $Ba_1(H_1, \Theta_1)$, assume $\text{Pa}(X_i(t))$ contains k_i variables X_{i1}, \dots, X_{ik_i} . For each X_i , we enumerate each conditional probability regarding $X_i(t)$ in Θ_1 as a triplet (z_j, \mathbf{y}_j, p_j) , with $z_j \in \{0,1\}$, $\mathbf{y}_j = y_{j1}y_{j2} \dots y_{jk_i} \in \{0,1\}^{k_i}$, $p_j = \Pr\{X_i(t) = z_j \mid \text{Pa}(X_i(t)) = \mathbf{y}_j\}$ and there are 2^{k_i+1} such triplets. The triplets are arranged such that the first 2^{k_i} of them have $z_j = 1$, and p_j 's are in ascending order. For every $j \leq 2^{k_i}$, define a sequence of symbols $\tilde{\mathbf{x}}_j = \tilde{x}_{j1}\tilde{x}_{j2} \dots \tilde{x}_{jk_i}$, where we choose the variable x_{jd} for symbol \tilde{x}_{jd} if $y_{jd} = 1$, and choose \bar{x}_{jd} (the negation of variable x_{jd}) for symbol \tilde{x}_{jd} if $y_{jd} = 0$.

Letting $l(i) = 2^{k_i} + 1$, we define the set of $l(i)$ Boolean functions for gene x_i in the PBN as $\Psi_i = \{\psi_1^{(i)}, \dots, \psi_{l(i)-1}^{(i)}, \psi_{l(i)}^{(i)}\}$, where

$$\psi_m^{(i)} = \tilde{\mathbf{x}}_m \vee \tilde{\mathbf{x}}_{m+1} \vee \dots \vee \tilde{\mathbf{x}}_{l(i)-1}, \text{ for } 1 \leq m \leq l(i) - 1 \quad (32)$$

is a disjunction of conjunctions, and $\psi_{l(i)}^{(i)}$ is a zero function. Define the corresponding function selection probabilities, $\alpha_1^{(i)} = p_1$, $\alpha_m^{(i)} = p_m - p_{m-1}$ for $1 < m \leq l(i) - 1$, and $\alpha_{l(i)}^{(i)} = 1 - p_{l(i)-1}$, it can be verified that

$$\begin{aligned} \Pr\{X_i(t) = 1 \mid \text{Pa}(X_i(t)) = \mathbf{y}_j\} &= \sum_{m=1}^{l(i)} \psi_m^{(i)}(\mathbf{y}_j) \alpha_j^{(i)} = \sum_{m=1}^j \psi_m^{(i)}(\mathbf{y}_j) \alpha_j^{(i)} \\ &= p_1 + \sum_{m=2}^j (p_m - p_{m-1}) = p_j. \end{aligned} \quad (33)$$

Therefore, a binary DBN can be represented as a PBN $G(V, \Psi, \alpha)$, where $\Psi = \{\Psi_1, \dots, \Psi_n\}$, $\alpha = \{\alpha_1, \dots, \alpha_n\}$, and $\alpha_i = (\alpha_1^{(i)}, \dots, \alpha_{l(i)}^{(i)})$. It should be noted that the mapping from a binary DBN to an independent PBN is not unique, and the above representation is one solution.

Summarizing subsections 3.3.1 and 3.3.2, we have the following theorem [15].

Theorem 3 Independent PBNs $G(V, \Psi, \alpha)$ and binary-valued DBNs (Ba_0, Ba_1) whose initial and transition BNs Ba_0 and Ba_1 are assumed to have only within and between consecutive slice connections, respectively, can represent the same joint distribution over their common variables.

3.4. Structural Analysis

Boolean models, like any other networks, have two issues of interest: Is the model robust? Is the model controllable? From the standpoint of system stability, we require the model be robust, namely, resistant to small changes in the network; from the standpoint of network intervention, we desire that the network be controllable, such that it will respond to certain perturbation. There needs to be a balance of the two properties. These two questions encourage researchers to do the following, (1) Find structural properties of the network that are related to robustness and controllability; (2) Seek ways to analyze the effect of perturbations and to design control techniques.

In 3.4.1, (1) is addressed. We review some structural measures of Boolean models that quantify the propagation of expression level change from one gene to others (or vice versa). In 3.4.2, (2) is partly addressed, where we review structural perturbations, and present a methodology that analyzes the perturbation on Boolean functions. Since the control techniques are out of the scope of this paper, interested readers can find more information in the review articles [16, 17].

3.4.1. Quantitative Measures of the Structure

In gene regulatory networks, the interactions among genes are reflected by two facts: the connections among genes, and the Boolean functions defined upon the connection. No matter it is the robustness or the controllability issue we are interested in, it all boils down to one central question: how a change in the expression level of one gene leads to changes in other genes in the network and

vice versa. Here, we introduce three measures of the structural properties that are related to the question: canalization, influence and sensitivity.

When a gene is regulated by several parent genes through function f , some parent genes can be more important in determining its value than others. An extreme case is canalizing function, in which one variable (canalizing variable) can determine the function output regardless of other variables.

Definition 7 [18] A Boolean function $f: \{0,1\}^n \rightarrow \{0,1\}$ is said to be canalizing if there exists an $i \in \{1, \dots, n\}$ and $u, v \in \{0,1\}$ such that for all $x_1, \dots, x_n \in \{0,1\}$, if $x_i = u$ then $f(x_1, \dots, x_n) = v$.

In gene regulatory networks, canalizing variables are also referred to as the master genes. Canalization is commonly observed in real organisms, and it plays an important role in the stability of genetic regulation, as discussed in [19, 20]. Mathematically, researchers have shown that canalization is associated with the stability of Boolean networks. For more theoretical work, see [21-23].

Other than canalization, the degree of gene-gene interaction can be described in more general terms, and we define two quantitative measures, influence and sensitivity, as follows.

Consider a Boolean function f with input variables x_1, \dots, x_n . Letting $\mathbf{x} = (x_1, \dots, x_n)$, we define the influence of a gene on the function f .

Definition 8 [2] The influence of a variable x_j on the Boolean function f is the expectation of the partial derivative with respect to the distribution $D(\mathbf{x})$,

$$I_j(f) = E_D \left[\frac{\partial f(\mathbf{x})}{\partial x_j} \right] = \Pr \left[\frac{\partial f(\mathbf{x})}{\partial x_j} = 1 \right] = \Pr \{ f(\mathbf{x}) \neq f(\mathbf{x}^{(j)}) \} \quad (34)$$

Note that the partial derivative of f with respect to x_i is

$$\frac{\partial f(\mathbf{x})}{\partial x_j} = |f(\mathbf{x}) - f(\mathbf{x}^{(j)})|, \quad (35)$$

in which $\mathbf{x}^{(j)} = (x_1, \dots, 1-x_j, \dots, x_n)$ (with x_j toggled).

In a BN, since each node x_i has one regulatory function f_i , so the influence of node x_j (assuming it regulates x_i) on x_i is $I_j(x_i) = I_j(f_i)$. In a PBN, let the set of regulating functions for x_i is $\psi_1^{(i)}, \dots, \psi_{l(i)}^{(i)}$, with function selection probabilities $\alpha_1^{(i)}, \dots, \alpha_{l(i)}^{(i)}$, the influence of gene x_j on x_i will be

$$I_j(x_i) = \sum_{k=1}^{l(i)} I_j(\psi_k^{(i)}) \cdot \alpha_k^{(i)}. \quad (36)$$

Thus for a Boolean model with n genes, an *influence matrix* Γ of dimension $n \times n$ can be constructed, where its i, j element being $\Gamma_{ij} = I_j(x_i)$. We can define influence of gene x_i to be the collective influence of x_i on all other genes,

$$r(x_i) = \sum_{j=1}^n \Gamma_{ij}. \quad (37)$$

Related to influence, we define the sensitivity of a function,

$$s_x(f) = \sum_{j=1}^n |f(\mathbf{x}) - f(\mathbf{x}^{(j)})|. \quad (38)$$

Then the average sensitivity of f with respect to distribution D is

$$s(f) = E_D[s_x(f)] = \sum_{j=1}^n E_D[|f(\mathbf{x}) - f(\mathbf{x}^{(j)})|] = \sum_{j=1}^n I_j(f). \quad (39)$$

The meaning of average sensitivity is that, on average, how much the function f changes between the Hamming distance one neighbors (i.e., the input vectors differ by one bit). For PBNs, the average sensitivity of gene x_i is (cf. Eq. (37))

$$s(x_i) = \sum_{j=1}^n I_j(x_i) = \sum_{j=1}^n \Gamma_{ji}. \quad (40)$$

Biologically, the influence of a gene indicates its overall impact on other genes. A gene with high influence has the potential to regulate the system dynamics and its perturbation has significant downstream effect. The sensitivity of a gene measures its stability or autonomy. Low sensitivity means that other genes have little effect on it, and the ‘‘house-keeping’’ genes usually have this property [2]. It is shown that such quantitative measures (or variants) can help guide the control of genetic networks [24] and aid in the steady-state analysis [25].

3.4.2. Structural Perturbation Analysis

There are two types of perturbation on Boolean models: perturbation on network states and perturbation on network structure. The former refers to a sudden (forced or spontaneous) change in the current state from \mathbf{x} to \mathbf{x}' , which causes the system dynamics to be disturbed temporarily. Such disturbance is transient in nature, because the network nodes and connections are intact, and the underlying gene regulation principles do not change. Therefore, the network attractors and the basins of attraction remain the same. However, if the perturbed Boolean model has multiple attractors, state perturbations may cause convergence to a different attractor than the original one, and may change the steady-state distribution of the network. This type of perturbation has been studied extensively (e.g. [26]), and finds its use in network control (e.g. [6]).

Perturbation on network structure refers to any change in the “wiring” or functions of the network. For instance, we may remove or add a gene to the network, change connections among genes, change the Boolean functions, or even change the synchronous Boolean network to an asynchronous model (where not all the genes are updated at the same time). Structural perturbation is more complex and less studied, compared to state perturbation. When network structure is perturbed, the network attractors and basins of attraction will be impacted, therefore the long-term consequence is more difficult to gauge than that of state perturbation.

The reasons for studying structural perturbation are: (1) modeling of gene regulatory networks is subject to uncertainty, and it is desirable to study the effect of small difference in network models on the network dynamic behavior; (2) it is likely that gene regulations, like other biological functions, have intrinsic stochasticity, and it is of interest to predict the consequence of any perturbation in regulation; (3) changing the network structure can alter the network steady-state distribution, thus structural perturbation can be an alternative way (with respect to state perturbation) of network control [25, 27, 28].

In [8], the authors developed theories to predict the impact of function perturbations on network dynamics and attractors, and main results are presented below. For more applications, see [28]. For further analysis in terms of steady-state distribution and application in network intervention, see [25].

Problem formulation. Given a Boolean network $\beta(V, \mathbf{f})$, $V = \{x_1, \dots, x_n\}$, $\mathbf{f} = (f_1, \dots, f_n)$, if one or more functions have one or more flips on their truth table outputs, we would like to predict the effect on state transitions and attractors.

Assume gene x_i has k_i regulators $x_{i1}, x_{i2}, \dots, x_{ik_i}$, then the truth table of f_i has 2^{k_i} rows, as is shown below. The input vector on row j will be denoted $\mathbf{a}_j^i \in \{0,1\}^{k_i}$, for instance, $\mathbf{a}_1^i = 00 \dots 0$. If we flip the output on row j , then we call it a one-bit function perturbation on f_i , and denote it $f_i^{(j)}$.

Row label	$x_{i1}x_{i2} \dots x_{ik_i}$	$f_i(\cdot)$
1	00...0	0
2	00...1	1
⋮	⋮	⋮
2^{k_i}	11...1	0

Any state transition $\mathbf{s} \rightarrow \mathbf{w}$ contains n mappings, $f_i : \mathbf{s} \rightarrow w_i$. We define $\text{In}_i(\mathbf{s}) = (s_{i1}, s_{i2}, \dots, s_{ik_i})$, which is a sub-vector of \mathbf{s} that corresponds to the regulators of x_i .

The following proposition and corollaries state the basic effects of one-bit function perturbation on the state

transitions and attractors. Proofs and extensions to two-bit perturbations can be found in [28].

Proposition 1 A state transition $\mathbf{s} \rightarrow \mathbf{w}$ is affected by one-bit perturbation $f_i \rightarrow f_i^{(j)}$ if and only if $\text{In}_i(\mathbf{s}) = \mathbf{a}_j^i$. If the state transition is affected, the new state transition will be $\mathbf{s} \rightarrow \mathbf{w}^{(i)}$, where $\mathbf{w}^{(i)}$ is defined to be the same as \mathbf{w} except the i -th digit is flipped.

Corollary 1 If x_i has k_i regulators, then the one-bit perturbation $f_i \rightarrow f_i^{(j)}$ will result in 2^{n-k_i} changed state transitions in the state transition diagram. This is equivalent to 2^{n-k_i} altered edges in the state transition diagram.

Corollary 2 (Invariant singleton attractor) Suppose state \mathbf{s} is a singleton attractor. It will no longer be a singleton attractor following the one-bit perturbation $f_i \rightarrow f_i^{(j)}$ if and only if $\text{In}_i(\mathbf{s}) = \mathbf{a}_j^i$.

Corollary 3 (Emerging singleton attractor) A non-singleton-attractor state \mathbf{s} becomes a singleton attractor as a result of the one-bit perturbation $f_i \rightarrow f_i^{(j)}$ if and only if the following are true: (1) $\text{In}_i(\mathbf{s}) = \mathbf{a}_j^i$, and (2) absent the perturbation, $\mathbf{s} \rightarrow \mathbf{s}^{(i)}$.

We use the following toy example to demonstrate the above results. From these results, more applications can be derived, such as controlling the network steady-state distribution through function perturbation, or identifying functional perturbation by observing phenotype changes [28].

Example 1 Consider a BN with $n = 3$ genes,

$$x_1(t+1) = x_3(t), \tag{41}$$

$$x_2(t+1) = 0, \tag{42}$$

$$x_3(t+1) = \bar{x}_1(t)x_2(t) + x_1(t)\bar{x}_2(t), \tag{43}$$

where the truth table of f_3 is shown below and the state transition diagram is shown in Fig. (2).

Row label	x_1	x_2	$f_3(\cdot)$
1	0	0	0
2	0	1	1
3	1	0	1
4	1	1	0

If a one-bit perturbation forces f_3 to become $f_3^{(3)}$, since $k_3 = 2$, 2 state transitions will be affected. By Proposition 1, states 100 and 101 no longer transit to 001 and 101 but to 000 and 100 respectively. Because of that, attractor cycle $\{001, 100\}$ will be affected. Moreover, Corollary 2 predicts that the singleton attractor 000 is robust to the perturbation while 101 is not. The predictions are confirmed by the new state transition diagram shown in Fig. (3).

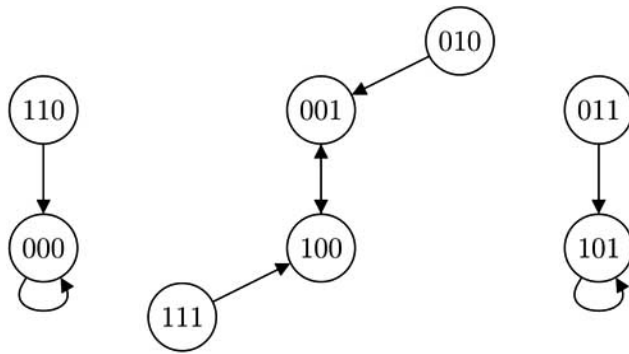


Fig. (2). State transition diagram of the original BN, Example 1.

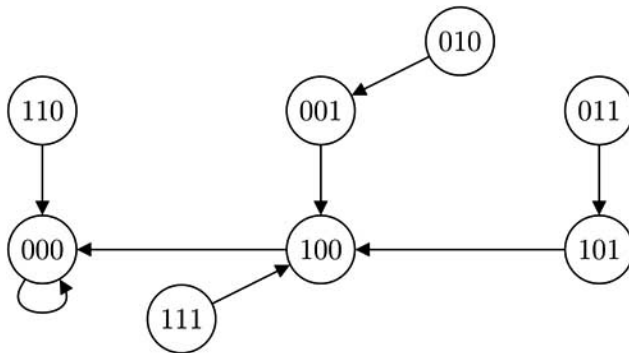


Fig. (3). State transition diagram of the perturbed BN, Example 1.

Finally, the author would like to remind the readers that other works on (various types of) structural perturbation are available. For instance, in [29], the authors added a redundant node to Boolean network, such that the bolstered network is more resistant to a one-bit function perturbation (as defined above). In [30], the effect of asynchronous update of a *drosophila* segment polarity network model is examined in terms of the phenotypes (steady-states). In [25], the authors derived analytical results of how function perturbations affects network steady-state distributions and applied them to structural intervention. In [31], the author modeled gene knockdown and broken regulatory pathway in Boolean networks, and analyzed the effects.

4. SIMULATION ISSUES WITH BOOLEAN MODELS

Recall from Section 2 that a Boolean model of n genes has a finite state space, and a BN has deterministic dynamic

behavior which can be fully captured by the state transition diagram. A PBN is probabilistic in nature, therefore its state transition is also probabilistic. For both BNs and PBNs, attractors are characteristic of their long-term behavior. Given the above knowledge, if we would like to know anything about a Boolean model, we should find out its state transition diagram and attractors first. This is to be discussed in Section 4.1.

For Boolean models, the most commonly encountered simulation issues include: (1) how to generate the time sequence data of a network, $\mathbf{x}(0), \mathbf{x}(1), \dots, \mathbf{x}(t), \dots$; (2) how to find the network steady-state distribution if it exists; and (3) how to produce artificial Boolean models with prescribed attractors to facilitate other studies. Among them, (1) is a basic practice that can be utilized in (2) and (3), and we will deal with them in Sections 4.2, 4.3 and 4.4 respectively. Note that the techniques in 4.1 is crucial to all the three issues.

4.1. Generating State Transition Diagram and Finding Attractors

To obtain the state transition diagram of a BN, we first compile a state transition table. Assuming n nodes in the network, x_1, \dots, x_n , we evaluate the current state $\mathbf{x}(t)$ to be $00\dots0, 00\dots1, \dots, 11\dots0$, and $11\dots1$ in turn, compute their respective $\mathbf{x}(t+1)$'s, and tabulate the results. The states can also be represented by integers instead of binary vectors. Table 1 is an example when $n = 3$. In practice, we only store the second row (“next states”) for computational purpose, because by default the current states are always arranged such that they correspond to integers $0, 1, 2, \dots, 2^n - 1$.

To obtain the state transition diagram of a BN, we draw 2^n vertices, each representing a possible state, and connect two vertices by a directed edge if one state transits to the other based on the state transition table. If a state transits to itself, the edge points to itself. Fig. (4) is the state transition diagram based on Table 1.

Similarly for a PBN, when gene perturbation rate $p = 0$, we can draw its state transition diagram by combining the state transition diagrams of its component BNs. Now each edge has a probability attached to it, representing the possibility of one state transiting to the other. For example, if a PBN is composed of two BNs, where the first BN has state transitions shown in Table 1, and the second BN has state

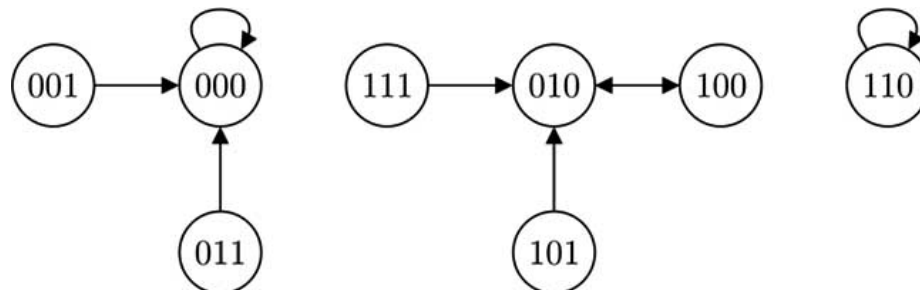


Fig. (4). State transition diagram of a Boolean network.

Table 1. Example of a State Transition Table for a BN

Current states	000	001	010	011	100	101	110	111
Next states	000	000	100	000	010	010	110	010

Table 2. State Transition Table for the Second BN in a PBN

Current states	000	001	010	011	100	101	110	111
Next states	001	001	101	001	010	010	110	010

transitions shown in Table 2, and their selection probabilities are c_1 and $c_2 = 1 - c_1$ respectively, then when $p = 0$, the PBN's state transition diagram is shown in Fig. (5). When $p > 0$, a state transition can either be driven by some network function or by random gene perturbations, and we may refer to its state transition matrix T when constructing the state transition diagram. It should be noted that the sum of probabilities of all the edges exiting a vertex should always be 1.

The following is a simple algorithm for finding the attractors of BN based on the state transition table (using integer representation of the states).

Algorithm 1 (Finding attractors)

1. Generate an array a of size 2^n , and initialize all a_i 's to 0. a_i corresponds to state $i - 1$.
2. Search for singleton attractors. For each state i between 0 and $2^n - 1$, look up the $(i + 1)$ -th entry in the state transition table for its next state j . If $j = i$, then j is a singleton attractor, set $a_{j+1} := 1$.
3. Search for attractor cycles. For each state i between 0 to $2^n - 1$, if $a_{i+1} = 0$, look up the state transition table repeatedly for the successor states of i , such that $i \rightarrow j \rightarrow k \dots$ until a singleton attractor or an attractor cycle is reached. If an attractor cycle is reached, save the cycle states and set the corresponding elements in a to 1.

4.2. Simulating a Dynamic System

A common practice with a Boolean model defined on $V = \{x_1, \dots, x_n\}$ is to generate time sequence data

$\mathbf{x}(0), \mathbf{x}(1), \mathbf{x}(2), \dots$. A direct method is to start from an initial state $\mathbf{x}(0)$, and plug in the Boolean functions repeatedly to find the subsequent states (for BNs and PBNs), sometimes taking into consideration network switches and gene perturbations (for PBNs).

An alternative way, which is more efficient when simulation time is long ($t \gg 2^n$), is to utilize the information of state transition diagram (encoded in the state transition table) or transition probability matrix T . For BN, it entails converting the current state $\mathbf{x}(t)$ to an integer, and looking up the state transition table or matrix T for the next state $\mathbf{x}(t + 1)$. For PBN, one can start from a randomly chosen initial state and a randomly chosen initial network (from r BNs), and follow either of the two protocols below. Note that we follow the notations in Definition 2 and use p, q to denote the random gene perturbation rate and network switching probability, respectively. Network selection probabilities are denoted by c_1, \dots, c_r .

- **Table-lookup and real-time computation based method.** Construct r state transition tables for the r component BNs respectively (letting gene perturbation rate $p = 0$). At any time t , if at the k -th network, generate n independent $[0, 1]$ uniformly distributed random numbers p_1, \dots, p_n . If $p_i < p$, flip $x_i(t)$ to get $x_i(t + 1)$; if $p_i > p, \forall i$ (no gene perturbation), convert $\mathbf{x}(t)$ to integer and look up the k -th state transition table to find $\mathbf{x}(t + 1)$. Finally, generate a $[0, 1]$ uniformly distributed random number q_s and compare to q to decide if the system will switch network at $t + 1$; if switch will occur,

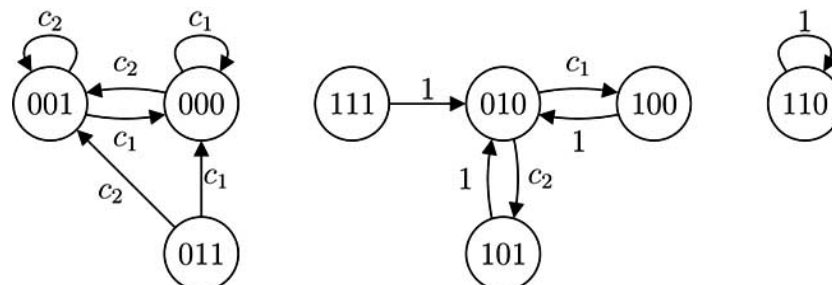


Fig. (5). Example of state transition diagram of a probabilistic Boolean network when gene perturbation rate $p = 0$.

choose from the r networks according to the selection probabilities.

- **T matrix based method.** Compute the transition probability matrix T . If $\text{dec}(\mathbf{x}(t)) = i-1$, generate a $[0,1]$ uniformly distributed random number p_t . If $\sum_{l=1}^{j-1} t_{il} \leq p_t < \sum_{l=1}^j t_{il}$ (t_{il} is the (i,l) element of T), then convert $j-1$ to a n -bit vector \mathbf{s} ($\text{dec}(\mathbf{s}) = j-1$) and the next state is $\mathbf{x}(t+1) = \mathbf{s}$.

One other issue of simulating a PBN is the choice of parameters p and q . As stated in Section 2, network switching probability q does not affect the probability of being at any constituent BN, and in theory we can choose any value for q ; however, we prefer to choose small q because in a biological system, switching network corresponds to the change of context (reflecting a change of regulatory paradigm, either caused by environment change or internal signals), which should not occur very often. Moreover, if q is large, or even $q=1$, then network switching is frequent, and a short time sequence of data $\mathbf{x}(t_1), \mathbf{x}(t_1+1), \dots, \mathbf{x}(t_2)$ are more likely to come from several BNs instead of from one single BN. This may pose a difficulty if we try to identify the underlying PBN and its component BNs from the sequence data [32]. On the other hand, if q is too small, and the number of BNs in the PBN is large, it will take too long a time to obtain the steady-state distribution by simulation method. Usually p should be small to reflect the rarity of random gene perturbation, and we let $p \ll q$. Also, small p is helpful if the generated sequence data will be used as artificial time-series data for the identification the underlying PBN and its component BNs. However, if p is too small, it will take longer to obtain the steady-state distribution. Usually, we can choose $q = 0.01 \sim 0.2$ and $p = 0.01\% \sim 0.5\%$.

4.3. Obtaining the Steady-State Distributions

4.3.1. Power Method

As discussed in Section 3.1, a PBN possesses a steady-state distribution when $q, p > 0$ [3]. By definition, this distribution π^* is the solution to linear equations $\pi = \pi \cdot T$ with constraint $\sum_i \pi_i = 1$, is unique and can be estimated by iteration, given the transition probability matrix T (assuming n genes, and $N = 2^n$).

Algorithm 2 (Finding steady-state distribution)

1. Set δ^* and generate an initial distribution $\pi^{(0)} = (\pi_1^{(0)}, \dots, \pi_N^{(0)})$; Let $k := 0$.
2. DO
Compute $\pi^{(k+1)} := \pi^{(k)} \cdot T$;
 $\delta := \|\pi^{(k+1)} - \pi^{(k)}\|$;

$k := k + 1$;

UNTIL ($\delta < \delta^*$)

3. $\pi^* := \pi^{(k)}$.

Note that $\|\cdot\|$ can be any norm, such as $\|\cdot\|_\infty$.

When the number of BNs in a PBN is large and some BNs have small selection probabilities, an approximation method for constructing T is proposed in [33]. In the approximation, \hat{T} is computed instead of T , which ignores r_0 BNs whose selection probabilities c_{k1}, \dots, c_{kr_0} are less than a threshold value ε ,

$$\hat{T} = \bar{T}' + \tilde{T}, \quad (44)$$

$$\bar{T}' = (1-p)^n \cdot \sum_{j=1, j \neq k1, \dots, kr_0}^r c_j T_j / \left(1 - \sum_{i=1}^{r_0} c_{ki} \right) \quad (45)$$

where T_j ($1 \leq j \leq r$) and \tilde{T} are defined as in Eqs. (7) and (8). If \hat{T} is used in place of T , and the solution for $\pi = \pi \hat{T}$ is $\hat{\pi}^*$, the expected relative error in steady-state distribution is shown to be bounded by $O(\varepsilon)$ [33]

$$E \left[\frac{\|\hat{\pi}^* - \hat{\pi}^* T\|_\infty}{\|\pi^*\|_\infty} \right] < (2 + 2n) \sum_{i=1}^{r_0} c_{ki} < 2(n+1)r_0\varepsilon. \quad (46)$$

The following is an alternative method of obtaining steady-state distribution. If we are interested in the attractors only, knowing that the majority of the steady-state probability mass is on attractors if p is small, we may apply the Markov chain based analytic method in Section 3.2.

4.3.2. Monte Carlo Simulation Method [34]

This method requires generation of a long time sequence of data, $\mathbf{x}(0), \mathbf{x}(1), \dots, \mathbf{x}(T)$, such that the frequencies of all the possible 2^n states approach the steady-state distribution. In a given n -gene PBN with gene perturbation rate p , the smaller p is and the larger n is, the longer it takes to converge to the steady-state distribution. In general, we need to simulate at least $10 \cdot 2^n \cdot p^{-1}$ steps.

To estimate when the PBN has converged to its steady-state distribution, we can use the Kolmogorov-Smirnov test. The basic idea of Kolmogorov-Smirnov test is to measure the closeness of an empirical probability distribution to the theoretical distribution. Since the latter (steady-state distribution) is unknown in this case, we will test the closeness of two empirical distributions.

To get two quasi-i.i.d (independently and identically distributed) samples in PBN, we select two samples $\mathbf{x}(t_1), \mathbf{x}(t_1 + m\Delta), \dots, \mathbf{x}(t_1 + (M-1)\Delta)$ and $\mathbf{x}(t_2), \mathbf{x}(t_2 + m\Delta), \dots, \mathbf{x}(t_2 + (M-1)\Delta)$ ($t_1 < t_2$ and $t_2 - t_1 \neq m\Delta, 0 < \forall m < M$), and the Kolmogorov-Smirnov statistic is defined as

$$K = \frac{1}{M} \max_{\mathbf{s}} \left| \sum_{m=0}^{M-1} \mathbf{1}_{[00\dots 0, \mathbf{s}]}(\mathbf{x}(t_1 + m\Delta)) - \sum_{m=0}^{M-1} \mathbf{1}_{[00\dots 0, \mathbf{s}]}(\mathbf{x}(t_2 + m\Delta)) \right|. \quad (47)$$

In the definition, the maximum is over the state space $\{0,1\}^n$, and $\mathbf{1}_{[00\dots 0, \mathbf{s}]}(\mathbf{x})$ is an indicator function whose output equals 1 if and only if $\mathbf{x} \in \{00\dots 0, \dots, \mathbf{s}\}$, $\mathbf{s} \in \{0,1\}^n$, and the output equals 0 otherwise.

4.4. Generating Artificial BNs with Prescribed Attractors [35]

In a simulation study of Boolean models, it is often necessary to create artificial networks with certain properties. Of special interest is the problem of generating artificial BNs with a given set of attractors, since attractors are hypothesized to correspond to cellular phenotypes and play an important role in the long term behavior of Boolean models.

First, note that the state transition diagram can be partitioned into level sets, where level set l_j consists of all states that transit to one of the attractors in exactly j steps, and the attractors belong to the level set l_0 .

Problem formulation [35] Given a set of n nodes $V = \{x_1, \dots, x_n\}$, a family of n subsets $P_1, \dots, P_n \subseteq V$ with $0 < k \leq |P_i| \leq K$, a set A of d states (binary vectors of n bits), and integers l, L satisfying $0 < l < L$, we will construct a BN defined on V , which satisfies the following constraints: the set of regulators of node x_i is P_i ($P = \{P_1, \dots, P_n\}$ is called the regulator set of V), the attractors are A_1, \dots, A_m such that $\cup_{j=1}^m A_j = A$, and the BN has between l and L level sets.

Specifically, if we are interested in constructing a BN with only singleton attractors, its state transition diagram will be a d -forest (containing d single-rooted trees) if the BN has d singleton attractors. The following theorem gives the number of all possible state transition diagrams that only contain singleton attractors (the proof can be found in [35]).

Theorem 4 The cardinality of the collection of all forests on N vertices is $(N+1)^{N-1}$.

Since $N = 2^n$ and the number of all possible state transition diagrams are N^N , when n is large, the ratio $(N+1)^{N-1}/N^N$ is asymptotically $e/2^n$, thus a brute force search has a low success rate.

Assuming only singleton attractors are allowed, the following algorithm is for solving the search problem formulated above. A second algorithm is also given in [35], but shown to be less efficient.

Algorithm 3 (Generating artificial Boolean network)

1. Randomly generate or give in advance a set A of d states (as singleton attractors).

2. Randomly generate a predictor set P , where each P_i has k to K nodes. If Step 2 has been repeated more than a pre-specified number of times, go back to Step 1.

3. Check if the attractor set A is compatible with P , i.e. only the attractors (each transits to itself) of the state transition diagram are checked for compatibility against P . If not compatible, go back to Step 2.

4. Fill in the entries of the truth tables that correspond to the attractors generated in Step 1. Using the predictor set P and randomly fill in the remaining entries of the truth table. If Step 4 has been repeated more than a pre-specified number of times go back to Step 2.

5. Search for cycles of any length in the state transition diagram D based on the truth table generated in Step 4. If a cycle is found go back to Step 4, otherwise continue to Step 6.

6. If D has less than l or more than L level sets go back to Step 4.

7. Save the generated BN and terminate the algorithm.

5. CLOSING WORDS

This paper has presented the following analysis and simulation issues of Boolean networks and probabilistic Boolean networks, which are models for gene regulatory networks.

- **Analysis.** An important aspect of Boolean models is that they can be viewed as homogeneous Markov chains; for a PBN, when the network switching probability $q > 0$ and gene perturbation rate $p > 0$, it possesses a steady-state distribution. Markov analysis serves as a basis for finding the steady-state probabilities of attractors and for proving the equivalence of PBN and dynamic Bayesian networks. Finally, a structural analysis is provided, where quantitative measures of gene-to-gene relationships are introduced, and the effect of perturbation on Boolean functions are analyzed.
- **Simulation.** Central to the simulation of Boolean models is the use of state transition diagram and transition probability matrix. In network simulation, different methods are presented and simple guidelines of parameter selection are provided. To test the convergence of a simulated PBN to its steady-state distribution, we can employ Kolmogorov-Smirnov statistic. Lastly, an algorithm for generating artificial BNs with prescribed attractors is presented.

To find more references on Boolean models, and obtain a MATLAB toolbox for BN/PBN, readers can go to the following website, <http://personal.systemsbiology.net/ilya/PBN/PBN.htm>. Another online source of papers is <http://gsp.tamu.edu/Publications/journal-publications>.

ACKNOWLEDGEMENT

The author thanks Yuping Xiao for giving valuable critique on the initial manuscript. The anonymous reviewers' insightful comments have helped the author's revision

considerably.

REFERENCES

- [1] Kauffman, S. A. *The origins of order: self-organization and selection in evolution*. Oxford University Press: New York; **1993**.
- [2] Shmulevich, I.; Dougherty, E. R.; Kim, S.; Zhang, W. Probabilistic Boolean networks: a rule-based uncertainty model for gene regulatory networks. *Bioinformatics*, **2002**, *18*, 261-274.
- [3] Shmulevich, I.; Dougherty, E. R.; Zhang, W. Gene perturbation and intervention in probabilistic Boolean networks. *Bioinformatics*, **2002**, *18*, 1319-1331.
- [4] Albert, R.; Othmer, H. G. The topology of the regulatory interactions predicts the expression pattern of the segment polarity genes in *Drosophila melanogaster*. *J. Theor. Biol.*, **2003**, *223*, 1-18.
- [5] Huang, S. Gene expression profiling, genetic networks, and cellular states: an integrating concept for tumorigenesis and drug discovery. *J. Mol. Med.*, **1999**, *77*, 469-480.
- [6] Datta, A.; Choudhary, A.; Bittner, M. L.; Dougherty, E. R. External control in Markovian genetic regulatory networks. *Mach. Learn.*, **2003**, *52*, 169-191.
- [7] Shmulevich, I.; Dougherty, E. R.; Zhang, W. From Boolean to probabilistic Boolean networks as models of genetic regulatory networks. *Proc. IEEE*, **2002**, *90*, 1778-1792.
- [8] Dougherty, E. R.; Xiao, Y. Design of probabilistic Boolean networks under the requirement of contextual data consistency. *IEEE Trans. Signal Processing*, **2006**, *54*, 3603-3613.
- [9] Zhang, S.-Q.; Ching, W.-K.; Ng, M. K.; Akutsu, T. Simulation study in probabilistic Boolean network models for genetic regulatory networks. *Int. J. Data Min. Bioinform.*, **2007**, *1*(3), 217-240.
- [10] Çinlar, E. *Introduction to stochastic processes*. 1st ed.; Prentice Hall: Englewood Cliffs, NJ; **1997**.
- [11] Brun, M.; Dougherty, E. R.; Shmulevich, I. Steady-state probabilities for attractors in probabilistic Boolean networks. *EURASIP J. Signal Processing*, **2005**, *85*, 1993-2013.
- [12] Friedman, N.; Linial, M.; Nachman, I.; Pe'er, D. Using Bayesian networks to analyze expression data. *J. Comput. Biol.*, **2000**, *7*, 601-620.
- [13] Ideker, T.; Thorsson, V. Discovery of regulatory interactions through perturbation: inference and experiment design. *Pac. Symp. Biocomput.*, **2000**, *5*, 302-313.
- [14] Xiao, Y.; Dougherty, E. R. Optimizing consistency-based design of context-sensitive gene regulatory networks. *IEEE Trans. Circ. Syst. I*, **2006**, *53*, 2431-2437.
- [15] Lähdesmäki, H.; Hautaniemi, S.; Shmulevich, I.; Yli-Harja, O. Relationship between probabilistic Boolean networks and dynamic Bayesian networks as models for gene regulatory networks, *Signal Processing*, **2006**, *86*, 814-834.
- [16] Datta, A.; Pal, R.; Dougherty, E. R. Intervention in probabilistic gene regulatory networks. *Curr. Genomics*, **2006**, *1*(2), 167-184.
- [17] Datta, A.; Pal, R.; Choudhary, A.; Dougherty, E. R. Control approaches for probabilistic gene regulatory networks, *IEEE Signal Processing Mag.*, **2007**, *24*(1), 54-63.
- [18] Shmulevich, I.; Lähdesmäki, H.; Dougherty, E. R.; Astola, J.; Zhang, W. The role of certain Post classes in Boolean network models of genetic networks, *PNAS*, **2003**, *100*(19), 10734-10739.
- [19] Szallasi, Z.; Liang, S. Modeling the normal and neoplastic cell cycle with realistic Boolean genetic networks: their application for understanding carcinogenesis and assessing therapeutic strategies. *Pac. Symp. Biocomput.*, **1998**, *3*, 66-76.
- [20] Martins-Jr, D. C.; Hashimoto, R. F.; Braga-Neto, U.; Bittner, M. L.; Dougherty, E. R. Intrinsically multivariate predictive genes. *IEEE J. Select. Topics Signal Processing*, **2008**, *2*, 424-439.
- [21] Kauffman, S.; Peterson, C.; Samuelsson, B.; Troein, C. Genetic networks with canalizing Boolean rules are always stable. *PNAS*, **2004**, *101*(49), 17102-17107.
- [22] Aldana, M.; Cluzel, P. A natural class of robust networks. *PNAS*, **2003**, *100*(15), 8710-8714.
- [23] Reichhardt, C. J. O.; Bassler, K. E. Canalization and symmetry in Boolean genetic regulatory networks. *Journal of Physics A: Math. Theor.*, **2007**, *40*, 4339-4350.
- [24] Choudhary, A.; Datta, A.; Bittner, M. L.; Dougherty, E. R. Assignment of terminal penalties in controlling genetic regulatory networks. In *American Control Conference*, Portland, OR, **2005**.
- [25] Qian, X.; Dougherty, E. R. Effect of function perturbation on the steady-state distribution of genetic regulatory networks: optimal structural intervention. *IEEE Trans. Signal Processing*, **2008**, *56*(10), 4966-4975.
- [26] Qu, X.; Aldana, M.; Kadanoff, L. P. Numerical and theoretical studies of noise effects in the Kauffman model. *J. Stat. Phys.*, **2002**, *109*(516), 967-985.
- [27] Shmulevich, I.; Dougherty, E. R.; Zhang, W. Control of stationary behavior in probabilistic Boolean networks by means of structural intervention. *Biol. Syst.*, **2002**, *10*, 431-445.
- [28] Xiao, Y.; Dougherty, E. R. The impact of function perturbations in Boolean networks. *Bioinformatics*, **2007**, *23*, 1265-1273.
- [29] Gershenson, C.; Kauffman, S. A.; Shmulevich, I. The role of redundancy in the robustness of random Boolean networks. *Artificial Life X, Proceedings of the Tenth International Conference on the Simulation and Synthesis of Living Systems*, **2006**, available online at <http://arxiv.org/abs/nlin/0511018>.
- [30] Chaves, M.; Albert, R.; Sontag, E. D. Robustness and fragility of Boolean models for genetic regulatory network. *J. Theor. Biol.*, **2005**, *235*, 431-449.
- [31] Xiao, Y. Modeling gene knockdown and pathway blockage in gene regulatory networks. In *Proc. IEEE Workshop on Systems Biology and Medicine*, Philadelphia, PA, November **2008**.
- [32] Marshall, S.; Yu, L.; Xiao, Y.; Dougherty, E. R. Inference of a probabilistic Boolean network from a single observed temporal sequence. *EURASIP J. Bioinform. Syst. Biol.*, **2007**, *1*, Article ID 32454, 15 pages.
- [33] Ching, W.-K.; Zhang, S.; Ng, M. K.; Akutsu, T. An approximation method for solving the steady-state probability distribution of probabilistic Boolean networks. *Bioinformatics*, **2007**, *23*, 1511-1518.
- [34] Shmulevich, I.; Gluhovsky, I.; Hashimoto, R.; Dougherty, E. R.; Zhang, W. Steady-state analysis of genetic regulatory networks modeled by probabilistic Boolean networks. *Comp. Funct. Genomics*, **2003**, *4*, 601-608.
- [35] Pal, R.; Ivanov, I.; Datta, A.; Bittner, M. L.; Dougherty, E. R. Generating Boolean networks with a prescribed attractor structure. *Bioinformatics*, **2005**, *21*, 4021-4025.